Journal of Computational and Applied Mathematics 235 (2011) 4047–4058

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A penalty function method based on smoothing lower order penalty function[☆]

Xinsheng Xu [a,*], Zhiqing Meng [b], Jianwu Sun [a], Rui Shen [b]

[a] Department of Mathematics and Information Science, Binzhou University, Binzhou256603, China

[b] College of Business and Administration, Zhejiang University of Technology, Hangzhou310023, China

ARTICLE INFO

Article history:
Received 11 January 2010
Received in revised form 7 November 2010

MSC:
primary 90C30
secondary 90C55

Keywords:
Constrained optimization
Penalty function
Smoothing technique

ABSTRACT

The paper introduces a smoothing technique for a lower order penalty function for constrained optimization problems (COP). It is proved that the optimal solution to the smoothed penalty optimization problem is a $\frac{\epsilon}{2}$-approximate optimal solution to the original optimization problem under some mild assumptions. Based on the smoothed penalty function, an algorithm for solving COP is proposed and some numerical examples are given.

1. Introduction

In this paper, the COP:

$$
\begin{aligned}
(\mathrm{P}) \quad & \min f(x) \\
& \text{s.t. } g_i(x) \le 0, \quad i = 1, 2, \ldots, m
\end{aligned}
$$

is taken into consideration, where f and $g_i : R^n \to R, i \in I = \{1, 2, \ldots, m\}$ are assumed to be twice continuously differentiable functions and $X = \{x \in R^n | g_i(x) \le 0,\ i = 1, 2, \ldots, m\}$ is the feasible set to (P). The penalty function method provides an efficient method in solving (P), which has received more and more attention from researchers [1–6]. Exact penalty functions are important, but some of the exact penalty functions are nondifferentiable which always prevent its efficient usage. Therefore, from the viewpoint of computation, many researchers paid more attention to smoothing a nondifferentiable penalty function [7–15]. Bertsekas [7] proposed a method of smoothing exact penalty functions. Zenios et al. [8] and Pinar and Zenios [9] gave a smooth exact penalty function for convex constrained optimization problems, which can be applied to obtain a good approximately optimal solution to (P). Chen and Mangasarian [10] obtained a smooth function to an approximate classical l_1 penalty function by consolidating the sigmoid function $\frac{1}{1+e^{-\alpha t}}$. Yang et al. [11] introduced a method of smoothing an exact penalty function. Meng et al. [12] proposed a method of smoothing a nonsmooth square-root exact penalty function for inequality constrained optimization problems. Liu [13] gave an approximation to smoothing the classical l_1 penalty function for nonlinear constrained optimization problems. However, the above literature

☆ This work is supported by the National Natural Science Foundation of China (10971193), the Natural Science Foundation of Zhejiang Province (Y6090063), the Natural Science Foundation of Shandong Province (ZR2010AM029) and the Science Foundation of Binzhou University (bzxykj0819).

* Corresponding author.
E-mail address: xxs0905@163.com (X. Xu).

0377-0427/$ – see front matter
doi:10.1016/j.cam.2011.02.031

paid little attention to smooth the lower order exact penalty function. In [14], Wu et al. discussed the lower order penalty function in the following form

$$F^k(x,\rho) = f(x) + \rho \sum_{i\in I} (\max\{g_i(x), 0\})^k \tag{1}$$

where $k \in (0, 1)$. Obviously, if $k = 1$, the lower order penalty function is reduced to the classical l_1 penalty function. Wu et al. [14] considered the ϵ-smoothing of (1) and got a modified exact penalty function under some mild assumptions for COP. Meng et al. [15] introduced a smoothing of a lower order penalty function and gave a robust SQP method by integrating the smoothed lower order penalty function with the SQP method. However, both the smoothed penalty functions proposed in [14,15] for lower order penalty functions is only first-order differentiable, which cannot be used in methods such as the Newton-type methods for solving COP. Therefore, it is essential to develop a smooth function which is at least twice continuously differentiable for the lower order exact penalty function.

In [8], Zenios et al. introduced a twice continuously differentiable function $\phi_2(\epsilon, t)$ to smooth the classical l_1 exact penalty function and got some useful results. Meng et al. [12] proposed a twice continuously differentiable function $q_\epsilon(t)$ to smooth the square-root exact penalty function for inequality constrained optimization problems. In this paper, following the ideas of [8,12], we construct a function $p_{\epsilon,k}(t)$ as follows:

$$p_{\epsilon,k}(t) = \begin{cases} 0 & t < 0, \\ \dfrac{m^2\rho^2 t^{3k}}{\epsilon^2} - \dfrac{m^3\rho^3 t^{4k}}{2\epsilon^3} & 0 \le t < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ t^k - \dfrac{\epsilon}{2m\rho} & t \ge \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}. \end{cases}$$

It is easy to prove that $p_{\epsilon,k}(t)$ is at least twice continuously differentiable on R. Using $p_{\epsilon,k}(t)$ as the smoothed function, a method for smoothing the lower order exact penalty function is introduced, and based on which an algorithm for solving COP is proposed herein.

The rest of this paper is organized as follows. Section 2 introduces a method of smoothing the lower order penalty function (1) in terms of second-order differentiability, with the relationships among the optimal objective function values of the smoothed penalty problem, the nonsmooth penalty problem and the original optimization problem being discussed. Based on the smoothed penalty function, Section 3 proposes an algorithm for solving COP. And in Section 4, some numerical examples using MATLAB are given, with conclusions given in Section 5.

2. A second-order smoothing function

Consider the function $p_k(t)$:

$$p_k(t) = \begin{cases} 0 & t < 0, \\ t^k & t \ge 0, \end{cases}$$

where $k \in (0, 1)$. Then, we can define the lower order penalty function for (P) as follows,

$$F^k(x,\rho) = f(x) + \rho \sum_{i\in I} p_k(g_i(x)), \tag{2}$$

where $\rho > 0$ is the penalty parameter. The corresponding penalty optimization problem for (P) is defined as

$$(\mathrm{P}_\rho) : \min F^k(x,\rho) \quad \text{s.t. } x \in R^n. \tag{3}$$

For $k \in (0, 1)$ and a given $\epsilon > 0$, the function $p_{\epsilon,k}(t)$ is defined as

$$p_{\epsilon,k}(t) = \begin{cases} 0 & t < 0, \\ \dfrac{m^2\rho^2 t^{3k}}{\epsilon^2} - \dfrac{m^3\rho^3 t^{4k}}{2\epsilon^3} & 0 \le t < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ t^k - \dfrac{\epsilon}{2m\rho} & t \ge \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \end{cases}$$

where ϵ is a smoothing parameter.

The function $p_{\epsilon,k}(t)$ has the following attractive properties:

Lemma 2.1. *For any $k \in (0, 1)$ and any $\epsilon > 0$, we have*

(i) *$p_{\epsilon,k}(t)$ is at least twice continuously differentiable on R, where*

$$p'_{\epsilon,k}(t) = \begin{cases} 0 & t < 0, \\ \dfrac{3km^2\rho^2 t^{3k-1}}{\epsilon^2} - \dfrac{2km^3\rho^3 t^{4k-1}}{\epsilon^3} & 0 \le t < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ kt^{k-1} & t \ge \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}} \end{cases}$$

and

$$p''_{\epsilon,k}(t) = \begin{cases} 0 & t < 0, \\ \dfrac{3k(3k-1)m^2\rho^2 t^{3k-2}}{\epsilon^2} - \dfrac{2k(4k-1)m^3\rho^3 t^{4k-2}}{\epsilon^3} & 0 \le t < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ k(k-1)t^{k-2} & t \ge \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}; \end{cases}$$

(ii) $\forall t \in R, p_k(t) \ge p_{\epsilon,k}(t)$;
(iii) $\lim_{\epsilon \to 0} p_{\epsilon,k}(t) = p_k(t)$.

Proof. (i) First, we will prove that $p_{\epsilon,k}(t)$ is continuous. Obviously, it is only necessary to prove the continuity of $p_{\epsilon,k}(t)$ at the separate points: 0 and $\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$.

(1) For $t = 0$, we have

$$\lim_{t\to 0^-} p_{\epsilon,k}(t) = \lim_{t\to 0^-} 0 = 0, \qquad \lim_{t\to 0^+} p_{\epsilon,k}(t) = \lim_{t\to 0^+} \left(\frac{m^2\rho^2 t^{3k}}{\epsilon^2} - \frac{m^3\rho^3 t^{4k}}{2\epsilon^3}\right) = 0,$$

which implies $\lim_{t\to 0^-} p_{\epsilon,k}(t) = \lim_{t\to 0^+} p_{\epsilon,k}(t) = 0 = p_{\epsilon,k}(0)$. Thus, $p_{\epsilon,k}(t)$ is continuous at $t = 0$.

(2) For $t = \left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$, we have

$$\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} p_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} \left(\frac{m^2\rho^2 t^{3k}}{\epsilon^2} - \frac{m^3\rho^3 t^{4k}}{2\epsilon^3}\right) = \frac{\epsilon}{2m\rho},$$

$$\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} p_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} \left(t^k - \frac{\epsilon}{2m\rho}\right) = \frac{\epsilon}{2m\rho},$$

which implies $\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} p_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} p_{\epsilon,k}(t) = \frac{\epsilon}{2m\rho} = p_{\epsilon,k}\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]$. Thus, $p_{\epsilon,k}(t)$ is continuous at $t = \left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$.

Then, we will prove $p_{\epsilon,k}(t)$ is first-order continuously differentiable, i.e. $p'_{\epsilon,k}(t)$ is continuous. Correspondingly, it is only necessary to prove the continuity of $p'_{\epsilon,k}(t)$ at the separate points: 0 and $\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$.

(1) For $t = 0$, we have

$$\lim_{t\to 0^-} p'_{\epsilon,k}(t) = \lim_{t\to 0^-} 0 = 0, \qquad \lim_{t\to 0^+} p'_{\epsilon,k}(t) = \lim_{t\to 0^+} \left(\frac{3km^2\rho^2 t^{3k-1}}{\epsilon^2} - \frac{2km^3\rho^3 t^{4k-1}}{\epsilon^3}\right) = 0,$$

which implies $\lim_{t\to 0^-} p'_{\epsilon,k}(t) = \lim_{t\to 0^+} p'_{\epsilon,k}(t) = 0 = p'_{\epsilon,k}(0)$. Thus, $p'_{\epsilon,k}(t)$ is continuous at $t = 0$.

(2) For $t = \left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$, we have

$$\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} p'_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} \left(\frac{3km^2\rho^2 t^{3k-1}}{\epsilon^2} - \frac{2km^3\rho^3 t^{4k-1}}{\epsilon^3}\right) = k\left(\frac{\epsilon}{m\rho}\right)^{\frac{k-1}{k}},$$

$$\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} p'_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} (kt^{k-1}) = k\left(\frac{\epsilon}{m\rho}\right)^{\frac{k-1}{k}},$$

which implies $\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} p'_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} p'_{\epsilon,k}(t) = k\left(\frac{\epsilon}{m\rho}\right)^{\frac{k-1}{k}} = p'_{\epsilon,k}\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]$. Thus, $p'_{\epsilon,k}(t)$ is continuous at $t = \left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$.

Finally, we will prove that $p''_{\epsilon,k}(t)$ is continuous. Correspondingly, it is only necessary to prove the continuity of $p''_{\epsilon,k}(t)$ at the separate points: 0 and $\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$.

(1) For $t = 0$, we have

$$\lim_{t\to 0^-} p''_{\epsilon,k}(t) = \lim_{t\to 0^-} 0 = 0, \qquad \lim_{t\to 0^+} p''_{\epsilon,k}(t) = \lim_{t\to 0^+}\left(\frac{3k(3k-1)m^2\rho^2t^{3k-2}}{\epsilon^2} - \frac{2k(4k-1)m^3\rho^3t^{4k-2}}{\epsilon^3}\right) = 0,$$

which implies $\lim_{t\to 0^-} p''_{\epsilon,k}(t) = \lim_{t\to 0^+} p''_{\epsilon,k}(t) = 0 = p''_{\epsilon,k}(0)$. Thus, $p''_{\epsilon,k}(t)$ is continuous at $t = 0$.

(2) For $t = \left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$, we have

$$\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} p''_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-}\left(\frac{3k(3k-1)m^2\rho^2t^{3k-2}}{\epsilon^2} - \frac{2k(4k-1)m^3\rho^3t^{4k-2}}{\epsilon^3}\right) = k(k-1)\left(\frac{\epsilon}{m\rho}\right)^{\frac{k-2}{k}},$$

$$\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} p''_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+}(k(k-1)t^{k-2}) = k(k-1)\left(\frac{\epsilon}{m\rho}\right)^{\frac{k-2}{k}},$$

which implies $\lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^-} p''_{\epsilon,k}(t) = \lim_{t\to\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]^+} p''_{\epsilon,k}(t) = k(k-1)\left(\frac{\epsilon}{m\rho}\right)^{\frac{k-2}{k}} = p''_{\epsilon,k}\left[\left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}\right]$. Thus, $p''_{\epsilon,k}(t)$ is continuous at $t = \left(\frac{\epsilon}{m\rho}\right)^{\frac{1}{k}}$.

(ii) For $\forall t \in R$, we have

$$p_k(t) - p_{\epsilon,k}(t) = \begin{cases} 0 & t < 0, \\ t^k - \left(\dfrac{m^2\rho^2t^{3k}}{\epsilon^2} - \dfrac{m^3\rho^3t^{4k}}{2\epsilon^3}\right) \geq \epsilon^2 & 0 \leq t < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ t^k - \left(t^k - \dfrac{\epsilon}{2m\rho}\right) = \dfrac{\epsilon}{2m\rho} & t \geq \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}. \end{cases}$$

That is $p_k(t) \geq p_{\epsilon,k}(t)$, $\forall t \in R$.

(iii) For $\forall t \in R$,

$$\lim_{\epsilon\to 0} p_{\epsilon,k}(t) = \begin{cases} = \lim\limits_{\epsilon\to 0} 0 = 0 = p_k(t) & t < 0, \\ = \lim\limits_{\epsilon\to 0}\left(\dfrac{m^2\rho^2t^{3k}}{\epsilon^2} - \dfrac{m^3\rho^3t^{4k}}{2\epsilon^3}\right) = 0 = p_k(t) & 0 \leq t < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ = \lim\limits_{\epsilon\to 0}\left(t^k - \dfrac{\epsilon}{2m\rho}\right) = t^k = p_k(t) & t \geq \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}. \end{cases}$$

The proof is completed. □

Assume that f and g_i $(i = 1, 2, \ldots, m)$ are twice continuously differentiable. Consider the penalty function for (P):

$$F^k(x, \rho, \epsilon) = f(x) + \rho\sum_{i\in I} p_{\epsilon,k}(g_i(x)). \tag{4}$$

Clearly, (4) is at least twice continuously differentiable at any $x \in R^n$, with which the following penalty optimization problem is obtained:

$$(\mathrm{PI}_\rho) : \min F^k(x, \rho, \epsilon) \quad \text{s.t.}\ x \in R^n.$$

Next, we will study the relationships between (P_ρ) and (PI_ρ).

Lemma 2.2. *For any $x \in R^n$, we have*

$$\lim_{\epsilon \to 0} F^k(x, \rho, \epsilon) = F^k(x, \rho).$$

Proof. For $x \in R^n$ and $i \in I$, from the definition of $p_k(t)$ and $p_{\epsilon,k}(t)$, we have

$$p_k(g_i(x)) - p_{\epsilon,k}(g_i(x)) = \begin{cases} 0 & g_i(x) < 0, \\ 0 \le [g_i(x)]^k - \left[\dfrac{m^2\rho^2[g_i(x)]^{3k}}{\epsilon^2} - \dfrac{m^3\rho^3[g_i(x)]^{4k}}{2\epsilon^3}\right] < \dfrac{\epsilon}{m\rho} & 0 \le g_i(x) < \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}, \\ 0 < g_i(x) - \left[g_i(x) - \dfrac{\epsilon}{2m\rho}\right] = \dfrac{\epsilon}{2m\rho} & g_i(x) \ge \left(\dfrac{\epsilon}{m\rho}\right)^{\frac{1}{k}}. \end{cases}$$

That is

$$0 \le p_k(g_i(x)) - p_{\epsilon,k}(g_i(x)) \le \frac{\epsilon}{2m\rho}, \quad i \in I.$$

Thus,

$$0 \le \sum_{i \in I} p_k(g_i(x)) - \sum_{i \in I} p_{\epsilon,k}(g_i(x)) \le \frac{\epsilon}{2\rho},$$

which implies

$$0 \le \rho \sum_{i \in I} p_k(g_i(x)) - \rho \sum_{i \in I} p_{\epsilon,k}(g_i(x)) \le \frac{\epsilon}{2}.$$

Therefore,

$$0 \le F^k(x, \rho) - F^k(x, \rho, \epsilon) \le \frac{\epsilon}{2},$$

that is

$$\lim_{\epsilon \to 0} F^k(x, \rho, \epsilon) = F^k(x, \rho).$$

The proof is completed. □

Lemma 2.2 points out that the gap between $F^k(x, \rho, \epsilon)$ and $F^k(x, \rho)$ can be arbitrarily small as long as the smoothing parameter ϵ is sufficiently small. A direct result of Lemma 2.2 is given as follows.

Theorem 2.1. *Let $\{\epsilon_j\} \to 0$ be a sequence of positive numbers and assume x^j a solution to $\min_{x \in R^n} F^k(x, \rho, \epsilon_j)$ for some given $\rho > 0$. Let x' be an accumulation point of the sequence $\{x^j\}$. Then, x' is an optimal solution to (P_ρ).*

By Theorem 2.1, we can get an approximately optimal solution to (P_ρ) by solving (PI_ρ).

Definition 2.1 (*[14]*)**.** A point x_ϵ is said to be an ϵ-approximate optimal solution to (P) if x_ϵ is feasible to (P) and

$$|f^* - f(x_\epsilon)| \le \epsilon,$$

where f^* is the optimal objective value of (P).

Theorem 2.2. *Let x^* be an optimal solution to (P_ρ) and x' be an optimal solution to (PI_ρ) for some $\rho > 0$ and $\epsilon > 0$. Then,*

$$\lim_{\epsilon \to 0} F^k(x', \rho, \epsilon) = F^k(x^*, \rho).$$

If both x^* and x' are feasible to (P), then

$$f(x') \le f(x^*) \le f(x') + \frac{\epsilon}{2}, \tag{5}$$

i.e., x' is an $\frac{\epsilon}{2}$-approximate optimal solution to (P_ρ).

Proof. From Lemma 2.2, we get

$$0 \le F^k(x^*, \rho) - F^k(x^*, \rho, \epsilon) \le \frac{\epsilon}{2},$$

$$0 \le F^k(x', \rho) - F^k(x', \rho, \epsilon) \le \frac{\epsilon}{2}.$$

Under the assumption that x^* is an optimal solution to (P_ρ) and x' is an optimal solution to (PI_ρ), we get

$$F^k(x^*,\rho) \le F^k(x',\rho),$$
$$F^k(x',\rho,\epsilon) \le F^k(x^*,\rho,\epsilon).$$

Hence

$$\begin{aligned}0 &\le F^k(x^*,\rho) - F^k(x^*,\rho,\epsilon)\\ &\le F^k(x^*,\rho) - F^k(x',\rho,\epsilon)\\ &\le F^k(x',\rho) - F^k(x',\rho,\epsilon)\\ &\le \frac{\epsilon}{2}.\end{aligned}$$

That is

$$\lim_{\epsilon\to 0} F^k(x',\rho,\epsilon) = F^k(x^*,\rho)$$

and

$$0 \le f(x^*) + \rho\sum_{i\in I} p_k(g_i(x^*)) - \left[f(x') + \rho\sum_{i\in I} p_{\epsilon,k}(g_i(x'))\right] \le \frac{\epsilon}{2}.$$

Furthermore, if x^* and x' are feasible to (P), then

$$\sum_{i\in I} p_k(g_i(x^*)) = \sum_{i\in I} p_{\epsilon,k}(g_i(x')) = 0.$$

Therefore,

$$0 \le f(x^*) - f(x') \le \frac{\epsilon}{2}.$$

That is

$$f(x') \le f(x^*) \le f(x') + \frac{\epsilon}{2}.$$

The proof is completed. □

By Theorem 2.2, we conclude that the gap between the optimal objective values of (P_ρ) and (PI_ρ) can be controlled through the smoothing parameter ϵ, if both x^* and x' are feasible to (P) and the optimal solution of (PI_ρ) is an $\frac{\epsilon}{2}$-approximate optimal solution of (P_ρ). So, we can get an $\frac{\epsilon}{2}$-approximate solution to (P_ρ) by solving (PI_ρ) under some mild conditions.

Definition 2.2. For $x' \in R^n$, we call $y' \in R^n$ a Lagrange multiplier vector corresponding to x' if and only if x' and y' satisfy

$$\nabla f(x') + \sum_{i\in I} y_i'\nabla g_i(x') = 0, \tag{6}$$

$$y_i' g_i(x') = 0, \quad y_i' \ge 0,\ g_i(x') \le 0,\ i = 1,2,\ldots,m. \tag{7}$$

Theorem 2.3. *Suppose that f and g_i $(i = 1,2,\ldots,m)$ are convex and let x^* be an optimal solution to* (P). *If x' is an optimal solution to* (PI_ρ) *and feasible to* (P), *and let $y' \in R^n$ be a Lagrange multiplier vector corresponding to x', then for any $\epsilon > 0$,*

$$f(x^*) \le f(x') \le f(x^*) + \frac{\epsilon}{2}. \tag{8}$$

i.e., x' is an $\frac{\epsilon}{2}$-approximate optimal solution to (P).

Proof. Since f and g_i $(i = 1,2,\ldots,m)$ are convex and continuously differentiable, we have

$$f(x^*) \ge f(x') + \nabla f(x')^T(x^* - x'), \tag{9}$$

$$g_i(x^*) \ge g_i(x') + \nabla g_i(x')^T(x^* - x'), \quad i = 1,2,\ldots,m. \tag{10}$$

Since x^* is an optimal solution to (P) and $y' \in R^n$ is a Lagrange multiplier vector corresponding to x', after applying (6), (7), (9) and (10), we have

$$\begin{aligned}F^k(x^*,\rho) &= f(x^*) + \rho\sum_{i\in I} p_k(g_i(x^*))\\ &\ge f(x') + \nabla f(x')^T(x^* - x')\end{aligned}$$

$$
\begin{aligned}
&= f(x') - \sum_{i\in I} y_i' \nabla g_i(x')^T (x^* - x') \\
&\geq f(x') - \sum_{i\in I} y_i' [g_i(x^*) - g_i(x')] \\
&= f(x') - \sum_{i\in I} y_i' g_i(x^*) \\
&\geq f(x').
\end{aligned}
$$

From Lemma 2.1, we have

$$0 \leq F^k(x^*, \rho) - F^k(x^*, \rho, \epsilon) \leq \frac{\epsilon}{2},$$

that is

$$
\begin{aligned}
f(x') &\leq F^k(x^*, \rho) \\
&\leq F^k(x^*, \rho, \epsilon) + \frac{\epsilon}{2} \\
&= f(x^*) + \rho \sum_{i\in I} p_{\epsilon,k}(g_i(x^*)) + \frac{\epsilon}{2} \\
&= f(x^*) + \frac{\epsilon}{2}.
\end{aligned}
$$

Since x' is feasible to (P), that is

$$f(x^*) \leq f(x').$$

Then, (8) holds, and the proof is completed. □

By Theorem 2.3, an optimal solution to (PI_ρ) becomes an $\frac{\epsilon}{2}$-approximate optimal solution to (P) under some mild conditions. Therefore, we can obtain an approximate optimal solution to (P) by solving (PI_ρ).

3. The smoothing penalty function algorithm

In this section, using the smoothing penalty function in Section 2, we propose an algorithm to solve COP.

Definition 3.1. A point $x_\epsilon \in X$ is an ϵ-feasible solution to (P) if $g_i(x) \leq \epsilon$, $\forall i \in I$.

Algorithm I.

Step 1: Choose $x^0 > 0$, $\epsilon > 0$, $\rho_0 > 0$, $N > 1$ and $j = 0$.
Step 2: Use x^j as the starting point to solve

$$\min_{x\in R^n} F^k(x, \rho_j, \epsilon). \tag{11}$$

Let x^{j+1} be the optimal solution obtained.
Step 3: If x^{j+1} is ϵ-feasible to (P), then stop. Otherwise, let $\rho_{j+1} = N\rho_j$ and $j = j + 1$, then go to Step 2.

Remark. In this algorithm, as $N > 1$ and $j \to \infty$, the sequence $\rho_j \to +\infty$.

For $x \in R^n$, we denote

$$
\begin{aligned}
&I^0(x) = \{i | g_i(x) = 0,\ i \in I\}, \\
&I^-(x) = \{i | g_i(x) < 0,\ i \in I\}, \\
&I^+(x) = \{i | g_i(x) > 0,\ i \in I\}, \\
&I^-_\rho(x) = \left\{ i | g_i(x) < \frac{\epsilon}{m\rho},\ i \in I \right\}, \\
&I^+_\rho(x) = \left\{ i | g_i(x) \geq \frac{\epsilon}{m\rho},\ i \in I \right\}.
\end{aligned}
$$

Then, the following result is obtained.

Theorem 3.1. *For any $k \in (0, 1)$, assume $\lim_{\|x\| \to \infty} f(x) = +\infty$ and let $\{x^j\}$ be the sequence generated by Algorithm I. If the sequence $\{F^k(x^j, \rho_j, \epsilon)\}$ is bounded, then,*

(i) *$\{x^j\}$ is bounded,*
(ii) *any limit point x^* of $\{x^j\}$ belongs to X*
(iii) *any limit point x^* of $\{x^j\}$ satisfies*

$$\nabla f(x^*) + \sum_{i \in I^0(x^*)} \lambda_i \nabla g_i(x^*) = 0, \tag{12}$$
$$g_i(x^*)\lambda_i = 0, \forall i \in I^0(x^*).$$

Proof. (i) By the assumptions, there exists some number L such that

$$F^k(x^j, \rho_j, \epsilon) \le L, \quad j = 0, 1, 2, \dots. \tag{13}$$

Suppose to the contrary that $\{x^j\}$ is unbounded and without loss of generality

$$\|x^j\| \to \infty, \quad j \to \infty.$$

Then, from (13), we obtain

$$L \ge F^k(x^j, \rho_j, \epsilon) = f(x^j) + \rho_j \sum_{i \in I} p_{\epsilon,k}(g_i(x^j)) \ge f(x^j), \quad j = 0, 1, 2, \dots,$$

which results in a contradiction since $\lim_{\|x\| \to \infty} f(x) = +\infty$.

(ii) Next, we will prove that any limit point x^* of $\{x^j\}$ belongs to X. Without loss of generality, we assume that

$$\lim_{j \to \infty} x^j = x^*.$$

Suppose to the contrary that $x^* \notin X$, then there exists some $i \in I$ such that $g_i(x^*) \ge \alpha > 0$. Note that

$$\begin{aligned} F^k(x^j, \rho_j, \epsilon) &= f(x^j) + \rho_j \sum_{i \in I} p_{\epsilon,k}(g_i(x^j)) \\ &= f(x^j) + \rho_j \sum_{i \in I^+_{\rho_j}(x^j)} p_{\epsilon,k}(g_i(x^j)) + \rho_j \sum_{i \in I^-_{\rho_j}(x^j)} p_{\epsilon,k}(g_i(x^j)) \\ &= f(x_j) + \rho_j \sum_{i \in I^+_{\rho_j}(x^j)} \left[(g_i(x^j))^k - \frac{\epsilon}{2m\rho_j} \right] + \rho_j \sum_{i \in I^-_{\rho_j}(x^j)} \left(\frac{m^2\rho^2[g_i(x^j)]^{3k}}{\epsilon^2} - \frac{m^3\rho^3[g_i(x^j)]^{4k}}{2\epsilon^3} \right). \end{aligned} \tag{14}$$

Since $g_i(x^*) \ge \alpha > 0$, then for any sufficiently large j, the set $\{i | g_i(x^*) \ge \alpha\}$ is not empty, then there exist a $i' \in I$ that satisfies $g_{i'}(x^*) \ge \alpha$. If $j \to \infty$, $\rho_j \to \infty$, it follows from (14) that

$$F^k(x^j, \rho_j, \epsilon) \to \infty,$$

which contradicts the assumption that $\{F^k(x^j, \rho_j, \epsilon)\}$ is bounded.

(iii) By Step 2 of Algorithm I, we have

$$\nabla F^k(x^j, \rho_j, \epsilon) = 0,$$

that is

$$\nabla f(x^j) + \rho_j \sum_{i \in I^+_{\rho_j}(x^j)} k(g_i(x^j))^{k-1} \nabla g_i(x^j) + \rho_j \sum_{i \in I^-_{\rho_j}(x^j)} \left[\frac{3km^2\rho_j^2(g_i(x^j))^{3k-1}}{\epsilon^2} - \frac{2km^3\rho_j^3(g_i(x^j))^{4k-1}}{\epsilon^3} \nabla g_i(x^j) \right] = 0,$$

which implies

$$\begin{aligned} &\nabla f(x^j) + \rho_j \sum_{i \in I^+_{\rho_j}(x^j)} k[p_k(g_i(x^j))]^{\frac{k-1}{k}} \nabla g_i(x^j) + \rho_j \sum_{i \in I^-_{\rho_j}(x^j)} \left[\frac{3km^2\rho_j^2[p_k(g_i(x^j))]^{\frac{3k-1}{k}}}{\epsilon^2} \right. \\ &\left. - \frac{2km^3\rho_j^3[p_k(g_i(x^j))]^{\frac{4k-1}{k}}}{\epsilon^3} \nabla g_i(x^j) \right] = 0. \end{aligned} \tag{15}$$

For $j = 0, 1, 2, \ldots$, let

$$\gamma_j = 1 + \sum_{i \in I^+_{\rho_j}(x^j)} \rho_j k [p_k(g_i(x^j))]^{\frac{k-1}{k}} + \sum_{i \in I^-_{\rho_j}(x^j)} \left[\frac{3km^2 \rho_j^2 [p_k(g_i(x^j))]^{\frac{3k-1}{k}}}{\epsilon^2} - \frac{2km^3 \rho_j^3 [p_k(g_i(x^j))]^{\frac{4k-1}{k}}}{\epsilon^3} \right].$$

Then, $\gamma_j > 1$, and it follows from (15) that

$$\frac{1}{\gamma_j} \nabla f(x^j) + \sum_{i \in I^+_{\rho_j}(x^j)} \frac{\rho_j k [p_k(g_i(x^j))]^{\frac{k-1}{k}}}{\gamma_j} \nabla g_i(x^j) + \sum_{i \in I^-_{\rho_j}(x^j)} \left[\frac{3km^2 \rho_j^3 [p_k(g_i(x^j))]^{\frac{3k-1}{k}}}{\gamma_j \epsilon^2} \right.$$
$$\left. - \frac{2km^3 \rho_j^4 [p_k(g_i(x^j))]^{\frac{4k-1}{k}}}{\gamma_j \epsilon^3} \right] \nabla g_i(x^j) = 0. \tag{16}$$

Let

$$\lambda^j = \frac{1}{\gamma_j},$$
$$\mu_i^j = \frac{\rho_j k [p_k(g_i(x^j))]^{\frac{k-1}{k}}}{\gamma_j}, \quad i \in I^+_{\rho_j}(x^j)$$
$$\mu_i^j = \left[\frac{3km^2 \rho_j^3 [p_k(g_i(x^j))]^{\frac{3k-1}{k}}}{\gamma_j \epsilon^2} - \frac{2km^3 \rho_j^4 [p_k(g_i(x^j))]^{\frac{4k-1}{k}}}{\gamma_j \epsilon^3} \right], \quad i \in I^-_{\rho_j}(x^j).$$

Then

$$\lambda^j + \sum_{i \in I} \mu_i^j = 1, \quad \mu_i^j \geq 0,\ i \in I,\ j = 0, 1, 2, \ldots. \tag{17}$$

As $j \to \infty$, we have

$$\lambda^j \to \lambda \geq 0, \tag{18}$$
$$\mu_i^j \to \mu_i \geq 0, \quad \forall i \in I. \tag{19}$$

From (16)–(19), we obtain

$$\lambda \nabla f(x^*) + \sum_{i \in I} \mu_i \nabla g_i(x^*) = 0,$$
$$\lambda + \sum_{i \in I} \mu_i = 1.$$

Let

$$\lambda_i = \frac{\mu_i}{\lambda},$$

we have

$$\nabla f(x^*) + \sum_{i \in (I^0(x^*) \bigcup I^-(x^*))} \lambda_i \nabla g_i(x^*) = 0$$

and

$$\lambda_i \geq 0, \quad \forall i \in I^0(x^*).$$

For $i \in I^-(x^*)$, as $j \to \infty$, we have

$$\mu_i = \lim_{j\to\infty} \mu_i^j \to= \lim_{j\to\infty} \left[\frac{3km^2 \rho_j^3 [p_k(g_i(x^j))]^{\frac{3k-1}{k}}}{\gamma_j \epsilon^2} - \frac{2km^3 \rho_j^4 [p_k(g_i(x^j))]^{\frac{4k-1}{k}}}{\gamma_j \epsilon^3} \right] = 0.$$

Therefore, $\mu_i = 0,\ \forall i \in I^-(x^*)$. Thus, (12) holds and the proof is completed. □

Theorem 3.1 points out that the sequence$\{x^j\}$ generated by Algorithm I may converge to a K–T point of (P) under some mild conditions.

Table 1
Results of Algorithm I with $k = \frac{1}{3}, x^0 = (0, 0, 0, 0), \rho_0 = 1, N = 10$ for (COP1).

No.iter j	Pen.par. ρ_j	Obj.fu.va. $f(x^j)$	App.sol. (x_1, x_2, x_3, x_4)
1	1	−77.662039	(1.779734, 1.862917, 4.630440, −2.778546)
2	10	−44.547342	(0.169754, 0.831694, 2.023476, −0.991323)
3	100	−44.237119	(0.169406, 0.835582, 2.008825, −0.965144)
4	1000	−44.233877	(0.170189, 0.835628, 2.008242, −0.965245)

Table 2
Results of Algorithms I, II and III with $x^0 = (0, 0, 0, 0), \rho_0 = 1, N = 10$ for (COP1).

Algorighm	No.iter j	Pen.par. ρ_j	Obj.fu.va. $f(x^j)$	App.sol. (x_1, x_2, x_3, x_4)
I	1	1	−77.662039	(1.779734, 1.862917, 4.630440, −2.778546)
	4	1 000	−44.233877	(0.170189, 0.835628, 2.008242, −0.965245)
II	1	1	−48.629509	(0.339654, 0.677748, 2.240736, −1.231420)
	4	1 000	−43.454571	(0.179481, 0.893194, 1.949962, −0.916729)
III	1	1	−46.204979	(0.192648, 0.844548, 2.108774, −1.076979)
	7	1 000 000	−44.233838	(0.169748, 0.836236, 2.008175, −0.965416)

4. Numerical experiments

In this section, we will give the numerical examples with Algorithm I using MATLAB. In order to compare the efficiency of Algorithm I with the algorithms based on the classical l_1 penalty function and l_2 penalty function, the Algorithms II and III are listed as follows:

Algorithm II (*or III*).

Step 1: Choose $x^0 > 0, \epsilon > 0, \rho_0 > 0, N > 1$ and $j = 0$.
Step 2: Use x^j as the starting point to solve

$$\min_{x \in R^n} F_1(x, \rho_j) = f(x) + \rho_j \sum_{i \in I} \max\{g_i(x), 0\}$$

$$\left(\text{or } \min_{x \in R^n} F_2(x, \rho_j) = f(x) + \rho_j \sum_{i \in I} \max\{g_i(x), 0\}^2 \right).$$

Let x^{j+1} be the optimal solution obtained.
Step 3: If x^{j+1} is ϵ-feasible to (P), then stop. Otherwise, let $\rho_{j+1} = N\rho_j$ and $j = j + 1$, then go to Step 2.

Example 1. Consider the problem in [12],

$$\begin{aligned}
&(\text{COP1}) \min f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \\
&\text{s.t. } g_1(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \le 0, \\
&g_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0, \\
&g_3(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0.
\end{aligned}$$

For $k = \frac{1}{3}$, let $x^0 = (0, 0, 0, 0)$, $\rho_0 = 1$, $N = 10$ and $\epsilon = 0.000001$. Numerical results of Algorithm I to solve (COP1) are given in Table 1, with the numerical results obtained by Algorithms II and III for (COP1) shown in Table 2. Table 1 shows the approximate solution $x^4 = (0.170189, 0.835628, 2.008242, -0.965245)$ obtained by Algorithm I is a feasible solution for $g_1(x^4) = -0.0000032361$, $g_2(x^4) = -0.000020291$ and $g_3(x^4) = -1.8830$. From Table 2, it is found that Algorithm I yields slightly better objective function values than Algorithms II and III.

Example 2. Consider the example in [9,12],

$$\begin{aligned}
&(\text{COP2}) \min f(x) = 10x_2 + 2x_3 + x_4 + 3x_5 + 4x_6 \\
&\text{s.t. } g_1(x) = x_1 + x_2 - 10 = 0, \\
&g_2(x) = -x_1 + x_3 + x_4 + x_5 = 0, \\
&g_3(x) = -x_2 - x_3 + x_5 + x_6 = 0,
\end{aligned}$$

Table 3
Results of Algorithms I, II and III with $x^0 = (0, 0, \dots, 0)$, $\rho_0 = 1000$, $N = 2$ for (COP2).

Algorighm	No.iter j	Pen.par. ρ_j	Obj.fu.va. $f(x^j)$	App.sol. (x_1, x_2, x_3, x_4)
I	1	1 000	18.038351	(1.692653, 1.234077, 0.201055, 0.493765, 1.003608, 0.447721)
	2	2 000	117.000079	(1.650676, 8.349322, 0.091780, 0.558895, 1.000001, 7.441101)
	3	4 000	117.000042	(1.650683, 8.349317, 0.091781, 0.558908, 0.999995, 7.441104)
	4	8 000	117.000004	(1.650682, 8.349318, 0.091775, 0.558907, 1.000000, 7.441092)
II	1	1 000	25.950527	(1.560720, 1.999444, −0.171976, 0.702805, 1.029784, 0.626970)
	3	4 000	123.918322	(1.608741, 8.391259, 0.971062, 0.626000, 0.011679, 9.350643)
III	1	1 000	55.710430	(−0.241427, 5.858919, −2.921650, 3.712867, −1.017742, 0.576225)
	13	4 096 000	117.000089	(1.657397, 8.342606, 0.109240, 0.548157, 0.999999, 7.451848)

Table 4
Results of Algorithms I, II and III with $x^0 = (2, 2, 2)$, $\rho_0 = 100$, $N = 10$ for (COP3).

Algorighm	No.iter j	Pen.par. ρ_j	Obj.fu.va. $f(x^j)$	App.sol. (x_1, x_2, x_3, x_4)
I	1	100	944.210629	(2.500212, 4.217963, 0.979881)
	3	10 000	944.215656	(2.500000, 4.220957, 0.966189)
II	1	100	947.775367	(2.501592, 3.469350, 2.593613)
	5	1 000 000	947.541190	(2.500000, 3.514028, 2.530140))
III	1	100	944.197815	(2.501028, 4.206208, 1.031285)
	6	10 000 000	944.215652	(2.500000, 4.221305, 0.964666)

$$
\begin{aligned}
& g_4(x) = 10x_1 - 2x_3 + 3x_4 - 2x_5 - 16 \le 0, \\
& g_5(x) = x_1 + 4x_3 + x_5 - 10 \le 0, \\
& 0 \le x_1 \le 12, \\
& 0 \le x_2 \le 18, \\
& 0 \le x_3 \le 5, \\
& 0 \le x_4 \le 12, \\
& 0 \le x_5 \le 1, \\
& 0 \le x_6 \le 16.
\end{aligned}
$$

For $k = \frac{1}{3}$, let $x^0 = (0, 0, \dots, 0)$, $\epsilon = 0.1$, $\rho_0 = 1000$, and $N = 2$. Algorithms I, II and III are adopted to solve (COP2), with numerical results shown in Table 3.

As shown in Table 3, the Algorithm I converges to a good numerical stability. Furthermore, with the same number of iterations, Algorithm I yields better objective function values than both Algorithms II and III.

Example 3. Consider the problem in [12],

$$
\begin{aligned}
& \text{(COP3)} \min f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \\
& \text{s.t. } g_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0, \\
& g_2(x) = (x_1 - 5)^2 + x_2^2 + x_3^2 - 25 = 0, \\
& g_3(x) = (x_1 - 5)^2 + (x_2 - 5)^2 + (x_3 - 5)^2 - 25 \le 0.
\end{aligned}
$$

For $k = \frac{1}{3}$, let $x^0 = (2, 2, 2)$, $\epsilon = 1$, $\rho_0 = 100$ and $N = 10$. Algorithms I, II and III are adopted to solve (COP3), with numerical results given in Table 4.

As shown in Table 4, it is found that Algorithm I converges slightly faster than Algorithms II and III, and yields a slightly better objective function value than Algorithms II and III.

Example 4. Consider the problem in [12],

$$
\text{(COP4)} \min f(x) = x_1^2 + x_2^2 + x_3^2
$$

Table 5
Results of Algorithms I and II with $x^0 = (1, 1, 1)$, $\rho_0 = 100$, $N = 2$ for (COP4)′.

m	Algorithm I			Algorithm II		
	No.iter j	Pen.par. ρ_j	Obj.fu.va. $f(x^j)$	No.iter j	Pen.par. ρ_j	Obj.fu.va. $f(x^j)$
1	4	800	5.334687	2	200	5.334688
10	3	400	5.334765	2	200	5.334690
100	2	200	5.334687	2	200	5.336118
1000	4	800	5.335900	4	800	7.900587
2000	1	100	5.336072	5	1600	7.396654

$$\text{s.t. } g(x) = x_1 + x_2 e^{x_3 t} - 2\sin(4t) \le 0, \quad t \in [0, 1].$$

We discretize $T = [0, 1]$ into m equal parts and a constraint is obtained at $t = \frac{i}{m}$, $i = 1, 2, \ldots, m$. Then (COP4) is transformed into (COP4)′ as follows:

$$(\text{COP4})' \ \min f(x) = x_1^2 + x_2^2 + x_3^2$$

$$\text{s.t. } g(x) = x_1 + x_2 e^{x_3 \frac{i}{m}} - 2\sin\left(4\frac{i}{m}\right) \le 0, \quad i = 1, 2, \ldots, m.$$

For $k = \frac{1}{3}$, let $x^0 = (1, 1, 1)$, $\epsilon = 0.1$, $\rho_0 = 100$ and $N = 2$. Algorithms I and II are adopted to solve (COP4)′, with numerical results given in Table 5.

As shown in Table 5, Algorithm I converges to a slightly better objective function value than Algorithm II when m increases.

From the above numerical results, the following conclusion is drawn: in general, Algorithm I converges to some stable results for some COPs.

5. Conclusions

The paper researches the lower order penalty function for solving COP. A smoothing penalty function is introduced to find an approximation to this. It is shown that under some mild conditions, an approximate $\frac{\epsilon}{2}$-optimal solution to the nonsmooth penalty problem by solving the smoothed penalty problem is obtained. It is proved that the optimal solution to the smoothed penalty problem is the approximate optimal solution to the original optimization problem under some mild conditions. Finally, from the numerical experiments it is found that the algorithm based on the smoothed penalty function is efficient for solving some COPs.

Acknowledgements

The authors would like to express their gratitude to anonymous referees' detailed comments and remarks that help us improve our presentation of this paper considerably.

References

[1] W.I. Zangwill, Nonlinear programming via penalty function, Management Sciences 13 (1967) 334–358.
[2] S.P. Han, O.L. Mangasarian, Exact penalty function in nonlinear programming, Mathematical Programming 17 (1979) 251–269.
[3] X.X. Huang, X.Q. Yang, Duality and exact penalization for vector optimization via augumented Lagrangian, Journal of Optimization Theory and Applications 111 (2001) 615–640.
[4] A.M. Rubinov, B.M. Glover, X.Q. Yang, Extended lagrange and penalty functions in continuous optimization, Optimization 46 (1999) 327–351.
[5] A.M. Rubinov, B.M. Glover, X.Q. Yang, Decreasing functions with applications to penalization, SIAM Journal on Optimization 10 (1999) 289–313.
[6] X.Q. Yang, X.X. Huang, A nonlinear Lagrangian approach to constrained optimization problems, SIAM Journal on Optimization 11 (2001) 1119–1144.
[7] D.P. Bertsekas, Nondifferentiable optimization via approximation, Mathematical Programming Studies 3 (1975) 1–25.
[8] S.A. Zenios, M.C. Pinar, R.S. Dembo, A smooth penalty function algorithm for network-structured problems, European Journal of Operational Research 64 (1993) 258–277.
[9] M.C. Pinar, S.A. Zenios, On smoothing exact penalty functions for convex constrained optimization, SIAM Journal on Optimization 4 (1994) 486–511.
[10] C.H. Chen, O.L. Mangasarian, Smoothing methods for convex inequalities and linear complementarity problems, Mathematical Programming 71 (1995) 51–69.
[11] X.Q. Yang, Z.Q. Meng, X.X. Huang, G.T.Y. Pong, Smoothing nonlinear penalty functions for constrained optimization, Numerical Functional Analysis and Optimization 24 (2003) 351–364.
[12] Z.Q. Meng, C.Y. Dang, X.Q. Yang, On the smoothing of the square-root exact penalty function for inequality constrained optimization, Computational Optimization and Applications 35 (2006) 375–398.
[13] B.Z. Liu, On smoothing exact penalty functions for nonlinear constrained optimization problems, Journal of Applied Mathematics and Computing 30 (2009) 259–270.
[14] Z.Y. Wu, F.S. Bai, X.Q. Yang, L.S. Zhang, An exact lower order penalty functions and its smoothing in nonlinear programming, Optimization 53 (2004) 51–68.
[15] K.W. Meng, S.J. Li, X.Q. Yang, A robust SQP method based on a smoothing lower order penalty function, Optimization 58 (2009) 22–38.